CLAIMS

1. A method of managing an original executable code forming a program intended to be downloaded into a reprogrammable on-board computer system such as a microprocessor card (CP), the said code possessing a cryptographic signature (SIGN) and being executable by the microprocessor of the on-board system after verification by the latter of the validity of the said signature, the said method comprising the steps consisting of at least:

- off card: - identifying a modified executable code (CI') corresponding to the original code, adapted to a predefined specific use; and - from variations between the data of the original code (CI) and the corresponding modified code (CI'), calculating a software component (CL) which, when it is applied to the original code, makes it possible to reconstruct the modified code; and - signing the said software component (CL);

- downloading the signed original code and the signed software component into the card;

- on card: - verifying the signatures (SIGN, SIGN') respectively of the original code (CI) and of the software component (CL); - applying the software component (CL) to the original code (CI) so as to reconstruct the modified code (CI') for its execution by the microprocessor.

2. A method according to claim 1, characterised in that the original executable code (CI) consists of an intermediate code, executable by the on-board system microprocessor by means of a virtual machine for interpreting this intermediate code.

3. A method according to claim 2, characterised in that the virtual machine is provided with an execution stack and in that the downloaded software component (CL), applied on card to the original intermediate code (CI), makes it possible to reconstruct a modified intermediate code (CI') a priori satisfying the verification criteria for the said intermediate code according to which the operands of each instruction of the said code belong to the data types manipulated by this instruction and, on each target switching instruction, the execution stack of the virtual machine is empty.

4. A method according to claim 3, characterised in that the modified intermediate code (CI') obtained by the application of the software component is verified, before its execution by the microprocessor by means of the virtual machine, according to a process verifying that the modified intermediate code (CI') satisfies the verification criteria.

5. A method according to claim 1 or 2, characterised in that the downloaded software component (CL), applied on card to the original code (CI), makes it possible to reconstruct a modified code so that its execution is more rapid compared with that of the original code.

6. A method according to claim 1 or 2, characterised in that the downloaded software component (CL), applied on card to the original code (CI), makes it possible to reconstruct a modified code so that it procures an optimisation in terms of size compared with the original code.